Proceedings of the First PhD Symposium on Sustainable Ultrascale Computing Systems (NESUS PhD 2016)
Timisoara, Romania

Jesus Carretero, Javier Garcia Blas
Dana Petcu
(Editors)

February 8-11, 2016

Work in progress about enhancing the programmability and energy efficiency of storage in HPC and cloud environments

PhD Student
Pablo Llopis
University Carlos III, Spain
pllopis@arcos.inf.uc3m.es

PhD Advisor
Javier Garcia Blas
University Carlos III, Spain
fjblas@arcos.inf.uc3m.es

PhD Advisor
Florin Isaila
University Carlos III, Spain
florin@arcos.inf.uc3m.es

Abstract

We present the work in progress for the PhD thesis titled "Enhancing the programmability and energy efficiency of storage in HPC and cloud environments". In this thesis, we focus on studying and optimizing data movement across different layers of the operating system's I/O stack. We study the power consumption during I/O-intensive workloads using sophisticated software and hardware instrumentation, collecting time series data from internal ATX power lines that feed every system component, and several run-time operating system metrics. Data exploration and data analysis reveal for each I/O access pattern various power and performance regimes. These regimes show how power is used by the system as data moved through the I/O stack. We use this knowledge to build I/O power models that are able to predict power consumption for different I/O workloads, and optimize the CPU device driver that manage performance states to obtain great power savings (over 30%). Finally, we develop new mechanisms and abstractions that allow co-located virtual machines to share data with each other more efficiently. Our virtualized data sharing solution reduces data movement among virtual domains, leading to energy savings I/O performance improvements.

Keywords NESUS, PhD Symposium, Energy Efficiency, I/O, Storage, Data movement, HPC, Cloud

I. Introduction

Modern scientific discoveries have been driven by an insatiable demand for high performance computing. However, as we progress on the road to Exascale systems, energy consumption becomes a primary obstacle in the design and maintenance of HPC facilities. A simple extrapolation shows that an Exascale platform based on the most energy efficient hardware currently available in the Green500 would consume 120 MW. However, the desirable goal has been set by the DOE to 20 MW [2]Actually, hardware vendors are already trying to provide more energy-efficient parts and software developers are gradually increasing power-awareness in the current software stack, from applications to operating systems.

Data movement has been identified as an extremely important challenge among many others on the way towards the Exascale computing [2]. As the power cost of computation decreases, the cost of data movement increasingly becomes a more relevant issue [1]. The low performance of the I/O operations continues to present a formidable obstacle to reaching Exascale computing in the future large-scale systems especially in I/O-intensive scientific domains and simulations. This issue triggers a special interest in optimizing storage systems in data centers, and motivates the need for more research to improve the energy efficiency of storage technologies. Therefore, a first step to develop I/O optimizations is to further understand how energy is consumed in the complete I/O stack.

We focus on gaining a clear understanding of how

power is used during I/O operations across the software stack, and using this knowledge to provide solutions that optimize energy utilization and I/O performance.

II. Thesis overview

The purpose of this section is to present an overview that provides an holistic description of the work introduced in this thesis. The contributions constitute work that studies and optimizes data movement across different levels of the operating system's I/O stack. More precisely, we propose contributions to the understanding and optimization of I/O power consumption that span from virtualized environments, through the operating system's I/O stack, and including low-level CPU device drivers, as depicted in Figure 1. Our contributions show that through the understanding of the different operating system layers and their interaction, it is possible to achieve coordinations that optimize the energy consumption and increase performance of I/O workloads.

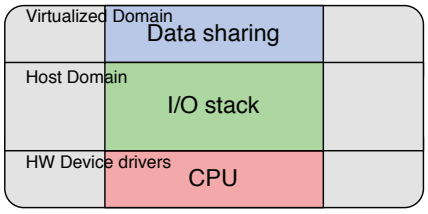

Figure 1: The contributions of this thesis span multiple levels of the software I/O stack.

The thesis starts with the goal of better understanding how power is used in the operating system's I/O stack. We perform a detailed study of power and energy usage across all across all system components during various I/O-intensive workloads [5]. To achieve an exhaustive examination, our work combines software and hardware-based instrumentation in order to study I/O data movement through exploratory data analysis. This data-driven process reveals detailed knowledge about how the system shifts between different power and performance regimes (depicted for a sequential file write in Figure 2), and which layers and algorithms of the I/O stack are responsible. As a result of our analysis and characterization, we provide I/O power models that are able to predict power consumption of I/O workloads that perform various access patterns. Figure 3 shows three workloads that do different combinations of random read/write, sequential read/write, strided reads combined re-reads (resulting in various page cache hit ratios). Our models are able to predict energy consumption with a normalized standard error under 5%.

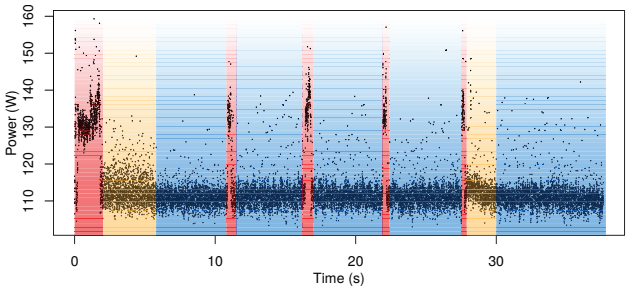

Figure 2: Power regimes during a sequential write of a 4 GiB file. Colors correspond to different regimes. Regimes correlate with speeds at which data is moved through the I/O stack, either put into the page cache or written to disk.

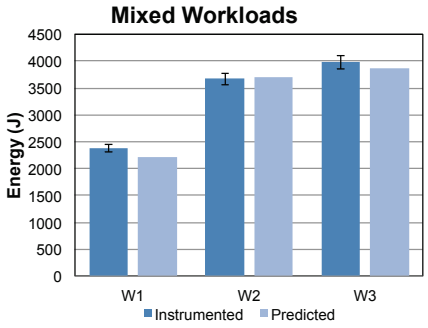

Figure 3: Comparison of measured energy with model predicted values for three workloads that mix reads and writes using different I/O patterns.

Our work continues into the hypervisor-based virtualization layer. We focus on optimizing data sharing between co-hosted virtual machines. In our work we refer to this as intra-domain data sharing, which mainly differs from existing solutions in the way the data moves across the software I/O stack. We develop virtualized data sharing (VIDAS) in order to

reduce data movement across virtual environments [6, 4]. VIDAS proposes new abstractions and mechanisms to more efficiently coordinate storage I/O across virtual domains, reduce data movement by creating intra-domain shared access spaces, relax POSIX consistency to allow flexible data write and update policies, and expose data locality. We argue that these abstractions and mechanisms can be used to build an efficient para-virtualized file system, and demonstrate reduced energy consumption and increased performance for various collective I/O access patterns. Figure 4 depicts the results for collectively writing and reading data to/from a 512MB object/file. The domains are accessing non-overlappingly interleaved strided vectors of 2MB blocks. Our solution uses a shared buffer space between domains/virtual machines, which reduces data movement. On the other hand, ROMIO collective operations copy the data into collective buffers before sending them to disks, which makes performance drop dramatically when increasing the number of virtual machines.

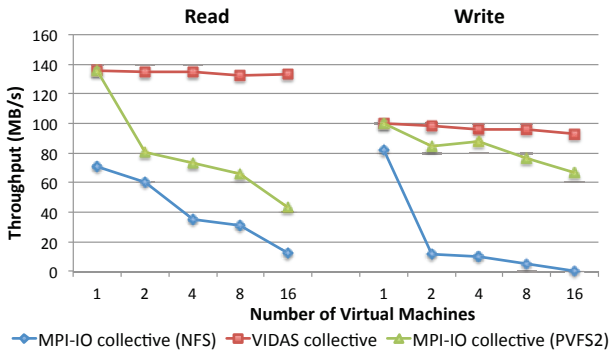

Figure 4: Comparison of VIDAS collective I/O and ROMIO collective I/O

Finally, we focus on the CPU, motivated by the fact that it is one of the most power-hungry components in a system. We examine the behavior of the CPU under I/O intensive workloads, and make two observations. First, we learn that in spite of being the most power-proportional component, the CPU does not shift performance states based on the I/O power and performance regimes revealed during our analysis of the operating system's I/O stack. Second, we note that there is a thermal imbalance that causes the CPU behave like a heterogeneous system. We develop kernel modules that use internal CPU mechanisms for thermal sensing and performance state selection, and demonstrate that we are able decrease energy consumption for I/O workloads for each of these two cases. Motivated by our first observation, we develop I/O-aware performance state selection. We are able to detecting I/O regimes and shift power states accordingly in order to lower CPU power usage without reducing performance. By adaptively setting performance states based on I/O performance regimes, we are able to reduce CPU energy consumption during write I/O by an average of 33%. Figure 5 depicts the difference between our solution and the Linux default CPU p-state driver in average CPU consumption, temperature (3.5°*C* improvement), and runtime (9% improvement).

Our second observation motivates us to develop thermal and I/O-aware thread placement, where computationally intensive and I/O intensive workload threads are placed in a thermal-aware fashion to optimize CPU power consumption. We are able to obtain up to 2.9% less energy consumption just by placing computation threads on the coldest CPU cores.

In conclusion, work shows that data movement within the host can be optimized to obtain performance and power consumption improvements. We not only analyze I/O power consumption in detail, but also demonstrate that data movement and I/O optimizations can be achieved on multiple layers of the system, spanning from the CPU device drivers, to virtual environments.

Acknowledgments

We would like to thank the community participating in this NESUS Action for making this PhD Symposium possible.

III. Related work

Our work is related to large body of research, but this Section will only highlight a few works. VIDAS builds upon and extends the paravirtualization concepts introduced first introduced Xen [8] to improve I/O performance in virtualized environments. Manousakis

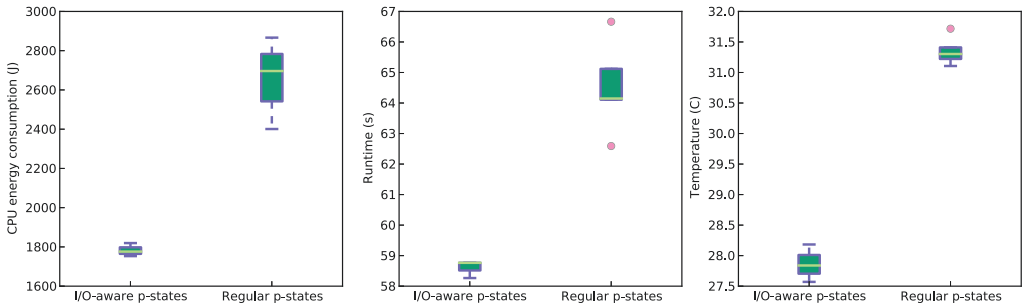

Figure 5: I/O-regime aware p-state selection driver consumes 33% less energy (left) than Intel's driver during write operations, takes 10% less time (middle), and decreases average CPU core temperature by 3.5° (right).

et al. [7] present a feedback-driven controller that improves DVFS for I/O intensive applications. They detect I/O phases and periodically switch the CPU frequency to all possible states, selecting the optimum setting power/performance ratio based on power readings from an internal power meter. Our solution does not rely on instrumented power readers, and detects power/performance regimes within I/O phases to shift p-states automatically. Our power meter instrument is based on the work provided in Powerpack [3]. Our CPU optimizations are also related to the work by [9], that addresses thermal variation and does thermal and workload-aware application placement.

References

[1] S. Borkar and A. A. Chien. The future of microprocessors. *Communications of the ACM*, 54(5):67–77, 2011.

[2] U. Department of Energy. Top Ten Exascale Research Challenges. Technical report, Department of Computer Science, Michigan State University, February 2014.

[3] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. W. Cameron. Powerpack: Energy profiling and analysis of high-performance systems and applications. *Parallel and Distributed Systems, IEEE Transactions on*, 21(5):658–671, 2010.

[4] P. Llopis, J. Blas, F. Isaila, and J. Carretero. Vidas: object-based virtualized data sharing for high performance storage i/o. In *Proceedings of the 4th ACM workshop on Scientific cloud computing*, pages 37–44. ACM, 2013.

[5] P. Llopis, M. F. Dolz, J. García-Blas, F. Isaila, J. Carretero, M. R. Heidari, and M. Kuhn. Analyzing power consumption of i/o operations in hpc applications. *Ultrascale Computing Systems (NESUS 2015) Krakow, Poland*, page 107, 2015.

[6] P. Llopis, G. Martin, B. Bergua, and J. Carretero. Virtual i/o forwarding for cloud-based hpc applications. In *Proceedings of the 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications*, pages 869–870. IEEE Computer Society, 2012.

[7] I. Manousakis, M. Marazakis, and A. Bilas. Fdio: A feedback driven controller for minimizing energy in i/o-intensive applications. In *Presented as part of the 5th USENIX Workshop on Hot Topics in Storage and File Systems, Berkeley, CA*, 2013.

[8] I. Pratt, K. Fraser, S. Hand, C. Limpach, A. Warfield, D. Magenheimer, J. Nakajima, and A. Mallick. Xen 3.0 and the art of virtualization. In *Linux Symposium*, page 65. Ottawa, Ontario, Canada, 2005.

[9] K. Zhang, S. Ogrenci-Memik, G. Memik, K. Yoshii, R. Sankaran, and P. Beckman. Minimizing thermal variation across system components. In *Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International*, pages 1139–1148. IEEE, 2015.